



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09244921 A**(43) Date of publication of application: **19.09.97**

(51) Int. Cl.

G06F 11/28(21) Application number: **08051516**(22) Date of filing: **08.03.96**(71) Applicant: **FUJITSU LTD**(72) Inventor:
OYAMA KAZUYA
KANEDA TOMOAKI
ISHIZAKI SHINICHI(54) **AUTOMATION SYSTEM FOR SOFTWARE TEST**

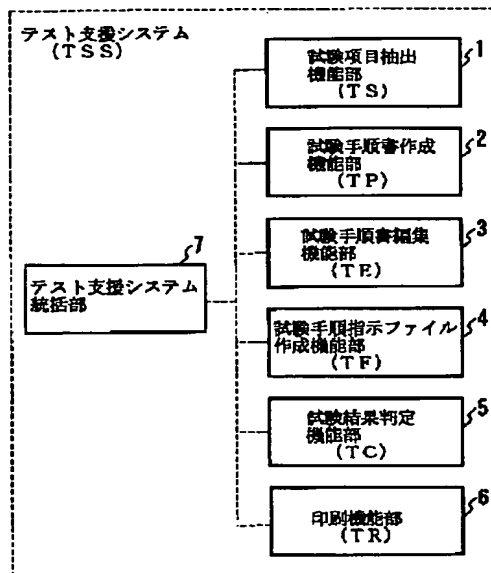
(57) Abstract:

PROBLEM TO BE SOLVED: To efficiently execute the test of a software and to improve software quality characteristics.

SOLUTION: A test item extraction function part 1 is provided and the test route and test item of a software test are automatically extracted from a detailed design book and a source file. A test procedure preparation function part 2 is provided and a test procedure model is automatically outputted based on a variable and a break point extracted from the detailed design book, the test route and the test item. Also, a test procedure editing function part 3 is provided, a set value and an expected value are inputted to the procedure model and a test procedure is automatically edited. A test procedure instruction file preparation function part 4 is provided and the test procedure instruction file corresponding to an optional debugger is automatically generated from the test procedure. Also, a tested result judgement function part 5 is provided, the result of the test is compared with the expected value, normal/defective conditions are automatically judged and the test procedure with the result is outputted. Also, a printing function part 6 is

provided and the test procedure and the test procedure with the result are routine-printed.

COPYRIGHT: (C)1997,JPO



Best Available Copy

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-244921

(43) 公開日 平成9年(1997)9月19日

(51) Int.Cl. ⁴	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/28	3 4 0		G 0 6 F 11/28	3 4 0 A

審査請求 未請求 請求項の数 4 O L (全 23 頁)

(21) 出願番号 特願平8-51516

(22) 出願日 平成8年(1996)3月8日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番1号

(72) 発明者 大山 和也

神奈川県横浜市港北区新横浜3丁目9番18号 富士通コミュニケーション・システムズ株式会社内

(72) 発明者 兼田 友晃

神奈川県横浜市港北区新横浜3丁目9番18号 富士通コミュニケーション・システムズ株式会社内

(74) 代理人 弁理士 柏谷 昭司 (外2名)

最終頁に続く

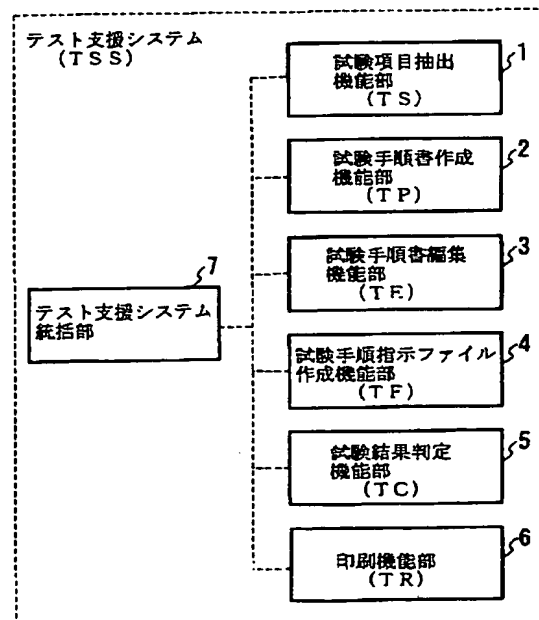
(54) 【発明の名称】 ソフトウェア試験の自動化システム

(57) 【要約】

【課題】ソフトウェア試験の自動化システムに関し、ソフトウェアの試験を効率的に実施し、かつ、ソフトウェア品質特性を向上させることを目的とする。

【解決手段】試験項目抽出機能部1を備えて、詳細設計書とソースファイルからソフトウェア試験の試験ルートと試験項目を自動的に抽出し、試験手順書作成機能部2を備えて、詳細設計書および試験ルートと試験項目から抽出した変数とブレークポイントをもとに試験手順書ひな形を自動的に出力し、試験手順書編集機能部3を備えて、手順書ひな形に設定値と期待値を入力して試験手順書を自動編集し、試験手順指示ファイル作成機能部4を備えて、試験手順書から任意のデバッグ対応の試験手順指示ファイルを自動生成し、試験結果判定機能部5を備えて、試験の結果を期待値と比較して良否の自動判定を行って結果付試験手順書を出し、印刷機能部6を備えて、試験手順書と結果付試験手順書を定型印刷する。

本発明の原理的構成を示す図



【特許請求の範囲】

【請求項 1】 ソフトウェアの詳細設計チャートをもとに、与えられた条件で動作するプログラム上のルートの組み合わせからなるすべての試験項目を自動的に抽出する試験項目抽出機能部と、ソースファイルを含む詳細設計チャートおよび前記抽出された試験項目をもとに試験ルートで参照する変数を自動的に抽出し、ソースファイルを含む詳細設計チャートをもとにブレークポイントを自動的に抽出するとともに、該変数とブレークポイントをもとに試験手順を手順書ひな形としてファイルおよび規定フォーマットに自動的に出力する試験手順書作成機能部と、前記手順書ひな形に設定値と期待値のデータまたはブレークポイントを入力するとともに、前記試験手順書作成機能によっては自動出力できないルート選択のための手順書コマンドを必要に応じて追加して試験手順書を自動編集する試験手順書編集機能部と、前記試験手順書をもとに、任意のデバッグに対応する試験用コマンドを自動作成して試験を実行し、試験対象となる上位モジュールが利用する下位モジュールを自動連結してボトムアップ試験を行うとともに、下位モジュールの試験方法を選択する試験手順指示ファイル作成機能部と、該試験結果を期待値および通過／非通過ブレークポイントと比較して良否の自動判定を行って結果付試験手順書を出力する試験結果判定機能部と、指定された試験手順書および結果付試験手順書のファイルを既存の印刷ツールを起動して定型印刷する印刷機能部とを備えたことを特徴とするソフトウェア試験の自動化システム。

【請求項 2】 前記試験項目抽出機能部で求めた試験項目によってスケジューリングして、前記試験手順書作成機能部と、試験手順書編集機能部と、試験手順指示ファイル作成機能部と、試験結果判定機能部と、印刷機能部の処理を連続して実施することを特徴とする請求項 1 に記載のソフトウェア試験の自動化システム。

【請求項 3】 請求項 1 または 2 に記載のソフトウェア試験の自動化システムにおいて、試験環境設定中および試験中の異常処理に対して、異常内容をメッセージとして自動出力する付加的機能を設けたことを特徴とするソフトウェア試験の自動化システム。

【請求項 4】 請求項 1 に記載のソフトウェア試験の自動化システムにおいて、前記各機能部の動作を統括するテスト支援システム統括部 7 において、以前使用した試験手順書と新たに作成した試験手順書とを比較して変更点がある部分以外の設定値情報を引き継ぐ処理を行うとともに、前記試験項目抽出機能部と、試験手順書作成機能部と、試験手順指示ファイル作成機能部と、試験結果判定機能部とにおける前回の状態をバックアップして復元可能とする差分情報収集部と、

指定されたデバッグ種別を認識してその条件に合致したデバッグコマンドを手順書言語に変換して、前記試験手順指示ファイルを試験手順書として生成するリバース変換部と、

前記試験結果判定機能部から出力された試験結果状況表をもとに、試験進捗状況を把握して試験のスケジューリングを行う試験状況管理部とを備えたことを特徴とするソフトウェア試験の自動化システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ソフトウェア試験を効率的に実施するとともに、ソフトウェアの品質向上を支援するための、ソフトウェア試験の自動化システムに関するものである。

【0002】ソフトウェアは、一般的にウォーターフォールモデルに従って開発される。これは、設計～製造行程の後に、単体、結合、総合の各試験を行なうことであって、ソフトウェアの品質特性（機能性、信頼性、使用性、効率性、保守性、移植性）を機能単位に確認してゆく作業である。

【0003】特に単体試験は、製造直後の試験であることから、その担う役割は大きい。単体試験は、各モジュール単位に、ホワイトボックス的な試験を実施する必要があるため、試験項目数が多く、試験工数も多くかかる。しかしながら、設計行程の遅れや、実機搬入時期、納期等によって、十分な工数を確保することができない場合が多い。

【0004】そのため、試験項目の抽出、手順書の作成、試験データの作成、試験結果の判定等が不十分なまま、試験が実施されてしまうことが多い。その結果、品質上の問題によって、後行程試験の工数を圧迫したり、運用後にソフトウェアの単純な不具合によって、システムが停止する等の問題を引き起こす場合もある。

【0005】そこで、ソフトウェアの製造直後の試験である単体試験を含めて、ソフトウェアの試験を効率的（期間、工数の大幅な削減）に実施し、かつ、ソフトウェア品質特性を向上させることが可能な、ソフトウェア試験の自動化システムが必要となる。

【0006】

【従来の技術】従来、ソフトウェアを自動的に試験する方法およびシステム（装置）に関しては、試験そのものを自動化することによって、試験の効率化（工数削減）およびソフトウェアの高信頼性化（結果の評価判定）を図る技術が、既に多数知られている（例えば、特開昭 58-205266 号「ソフトウェア自動試験方法」、特開昭 63-8948 号「交換ソフトウェアの自動試験方式」、特公平 5-16220 号「タスクモジュール自動検証方式」、特開平 3-38956 号「自動試験装置」、特開平 3-156645 号「ソフトウェア自動試験装置」、特開平 3-179963 号「情報処理装置会

話型試験システム」、特開平3-198457号「交換処理用呼処理プログラムの自動試験装置」、特開平5-244272号「交換ソフトウェアの自動試験方式」、特開平5-233349号「プログラムモジュールの自動試験ツール」、特開平5-282176号「自動試験実行装置」、特開平6-208484号「交換ソフトウェアの自動試験方式」、特開平7-93185号「ソフトウェア自動試験システム」、特開平7-200355号「ソフトウェア自動試験装置」)。

【0007】しかしながら、試験そのものを自動化しただけでは、真の意味の試験効率化には結びつかない。例えば、試験者は、自分が作成したプログラムが動作するように試験項目を設定したり、その項目に従って手順書や試験データを作成したりする処理を、人手で行なっているため、試験以前に、人為的なミスが入り込むことが多い。

【0008】特に、試験の目的を明確に定義して、試験を実施するのでなければ、試験が無意味なものになる。

【0009】また、試験結果を自動的に判定する技術は、試験結果の可否を試験者に通知するだけであって、プログラムのどの箇所に不具合があるのかは、試験者自身が判断しなければならない。

【0010】

【発明が解決しようとする課題】従来のソフトウェア試験方法のように、試験者が人手によって試験項目を抽出したり、試験手順書を作成したりするのでは、人為的なミスの発生を完全に防止することはできない。

【0011】従って、ソフトウェア試験そのものを効率化することができたとしても、上記の問題を解決しなければ、ソフトウェアの品質向上は望めない。

【0012】本発明は、このような従来技術の課題を解決しようとするものであって、ソフトウェア試験項目の抽出から、試験結果の判定までを、一連の作業として自動化するとともに、各試験者が実施した試験項目を自動集計し、予実管理までを行なうことによって、試験の効率化およびソフトウェア品質特性の確保を図ることが可能な、ソフトウェア試験の自動化システムを提供することを目的としている。

【0013】

【課題を解決するための手段】図1は、本発明の原理的構成を示したものであって、試験項目抽出から試験結果判定までの一連の作業を自動化した“ソフトウェア試験の自動化システム”の一例としての、テスト支援システム(TSS)の構成例を示したものである。

【0014】図1において、1は詳細設計書またはソース上から、試験項目を自動生成する試験項目抽出機能部(TS)、2は詳細設計書および試験項目抽出機能部1が生成した試験項目に基づいて、試験手順書を自動作成する試験手順書作成機能部(TP)、3は試験手順書作成機能部2が生成した試験手順書をもとに任意な編集を

可能にし、試験手順書記入もれ等のメッセージを通知する試験手順書編集機能部(TE)、4は試験手順書編集機能部3が編集した試験手順書から、任意のデバッグに対応した試験手順指示ファイルを自動生成する試験手順指示ファイル作成機能部(TF)、5は試験手順指示ファイル作成機能部4がと作成した試験手順指示ファイルをデバッグ上で実行して、得られた試験結果ファイルから試験結果を判定する試験結果判定機能部(TC)、6は各機能部が生成した文書(またはデータ)を印刷し、またはファイル化する印刷機能部(TR)である。

【0015】また7は、テスト支援システム統括部であって、各機能部の動作環境の設定を行なうとともに、各機能部の全体を統括管理する。試験項目抽出機能部1、試験手順書作成機能部2、試験手順書編集機能部3、試験手順指示ファイル作成機能部4、試験結果判定機能部5、印刷機能部6は、それぞれ単独にツールとして動作することができるとともに、テスト支援システム統括部7の統括のもとに、個別の組み合わせで、動作することができる。従って実施する試験項目をスケジューリングして、試験項目抽出機能部1から印刷機能部6までの機能を、連続して実施することも可能である。

【0016】図2、図3は、本発明の作用を説明する図(1/2)、(2/2)であって、従来の試験作業項目と本発明のソフトウェア試験の自動化システムでの作業項目との比較を示している。図示のように、本システムによって、項目セレクト、手順書作成と編集、コマンドファイル作成、デバッグ、結果判定を繰り返して実施することによって、従来、手作業で行なっていた各作業も含めて、試験の一連の作業が自動化され、従来技術の課題が解決されて、ソフトウェア試験の効率化およびソフトウェアの品質向上を図ることができるようになる。

【0017】以下、本発明の課題を解決するための具体的手段を掲げる。

【0018】(1) 本発明のソフトウェア試験の自動化システムは、ソフトウェアの詳細設計チャート(またはソースファイル)をもとに、与えられた条件で動作するプログラム上のルートの組み合わせからなるすべての試験項目を自動的に抽出する試験項目抽出機能部1と、詳細設計チャートおよび抽出された試験項目をもとに試験ルートで参照する変数を自動的に抽出し、詳細設計チャートをもとにブレイクポイントを自動的に抽出するとともに、この変数とブレイクポイントをもとに試験手順を手順書ひな形としてファイルおよび規定フォーマットに自動的に出力する試験手順書作成機能部2と、手順書ひな形に設定値と期待値のデータまたはブレイクポイントを入力するとともに、試験手順書作成機能では自動出力されないルート選択のための手順書コマンドを必要に応じて追加して試験手順書を自動編集する試験手順書編集機能部3と、試験手順書をもとに、任意のデバッグに対応する試験用コマンドを自動作成して試験を実行し、試験

対象となる上位モジュールが利用する下位モジュールを自動連結してボトムアップ試験を行うとともに、下位モジュールの試験方法を選択する試験手順指示ファイル作成機能部4と、試験結果を期待値と比較して良否の自動判定を行って結果付試験手順書出力する試験結果判定機能部5と、指定された試験手順書および結果付試験手順書のファイルを既存の印刷ツールを起動して定型印刷する印刷機能部6とを備えて構成される。

【0019】(2) (1) の場合に、試験項目抽出機能部1で求めた試験項目によってスケジューリングして、試験手順書作成機能部2と、試験手順書編集機能部3と、試験手順指示ファイル作成機能部4と、試験結果判定機能部5と、印刷機能部6の処理を連続して実施する。

【0020】(3) (1) または(2) の場合に、試験項目抽出機能部1、試験手順書作成機能部2、試験手順書編集機能部3、試験手順指示ファイル作成機能部4、試験結果判定機能部5、印刷機能部6、テスト支援システム統括部7の試験環境設定中および試験中の異常処理に対して、異常内容をメッセージとして自動出力する付加的機能を設ける。

【0021】(4) (1) の場合に、各機能部の動作を統括するテスト支援システム統括部7において、以前使用した試験手順書と新たに作成した試験手順書とを比較して変更点がある部分以外の設定値情報を引き継ぐ処理を行うとともに、試験項目抽出機能部1と、試験手順書作成機能部2と、試験手順指示ファイル作成機能部4と、試験結果判定機能部5における前回の状態をバックアップしてユーザ指定された部分または前回と今回との相違点を抽出してその情報を反映または復元できる差分情報収集部72と、指定されたデバッグ種別を認識してその条件に合致したデバッグコマンドを手順書言語に変換して、試験手順指示ファイルを試験手順書として生成する等本来の手順を逆に動作させるリバース変換部73と、試験結果判定機能部5から出力された試験結果状況表をもとに、試験進捗状況を把握して試験のスケジューリングを行う試験状況管理部74を備える。

【0022】

【発明の実施の形態】以下、本発明の一実施形態を、図面に基いて説明する。

〔1〕試験項目抽出機能部（TS）

試験項目抽出機能部は、詳細設計チャートまたはソースファイルを基に、試験対象となるルート（ある条件で動作するプログラムの道筋）を組み合わせ、すべての試験項目を自動抽出するものである。すなわち、試験項目を、試験対象となる詳細設計書やソースファイルを参照して解析して、その設計書またはソース上の“条件式”や“条件文”等の分岐情報を抽出し、これらの条件の組み合わせによって、その試験対象となる関数の試験項目を自動抽出して、指定されたファイルや規定フォーマット上に出力する。なお、この試験項目の抽出後、試験手

順書編集機能部等を利用することによって、ユーザにより任意編集（削除や追加、変更）が可能である。

【0023】試験項目抽出機能部は、以下の各機能ブロックから構成されている。

- ・分岐位置判定部
- ・分岐情報収集部
- ・試験項目出力部

【0024】図4は、試験項目抽出機能部の入出力を示したものであって、試験項目抽出機能部1に対して、詳細設計書とソースファイルのいずれか一方または両方を入力して、試験ルート表と試験項目表のいずれか一方または両方を出力できることが示されている。

【0025】図5は、試験項目抽出機能部の詳細機能の構成例を示すブロック図であって、試験項目抽出機能部メイン10のもとに、分岐位置判定部11、分岐情報収集部12、試験項目出力部13を有することが示されている。

【0026】図6は、試験項目抽出機能部の動作フローの概略を示したものであって、試験ルートおよび試験内容の生成フローの一例を示す。図中、◇は判定文を示し、□は処理を示している。

【0027】以下、分岐情報収集部で行なう情報収集要素、および試験項目出力部で行なう情報出力要素について説明する。情報出力要素には試験項目表と試験ルート表の2通りがあり、任意に規定ファイルや規定フォーマットに出力可能である。これらを生成するために必要な情報要素は、以下のようなものである。

【0028】(1) 試験項目表

試験項目表については、以下の項目を任意に抽出して出力する。

1. 試験対象関数名（英字名、日本語名、コメント等）
2. 試験項目番号
3. 試験項目名
4. 試験内容
5. 試験データ名
6. 試験項目一覧表作成日
7. 試験項目一覧表作成者名
8. その他（備考欄）

【0029】図7は、試験項目一覧表の出力例を示したものである。試験（テスト）項目一覧表として、テスト項目名、テスト内容、テストデータ名、備考が出力されることが示されている。

【0030】(2) 試験ルート表

試験ルート表については、以下の項目を任意に抽出して出力する。

1. 試験対象関数名（英字名、日本語名、コメント等）
2. 試験項目番号
3. 試験項目名
4. 試験内容
5. 試験データ名

6. 試験ルート表作成日
7. 分岐点の内容
8. 分岐点の通過・非通過のデータ
9. 条件文の記述位置（ソース上行番号等）
10. その他（備考欄）

【0031】図8は、試験ルート表の出力例を示したものである。試験（テスト）ルート表として、テスト項目番号、テスト項目名が条件式とともに、行番号ごとに出力されることが示されている。なお、図中において二重線枠内は、試験ルート情報であって、真偽の表記がない場合は、そのルートを通らない。またIF文でELSEがない場合でも、真でなければ、偽の表記がなされる。

【0032】〔2〕試験手順書作成機能部（TP）
試験手順書作成機能部は、詳細設計チャート（またはソースファイル）および試験項目を基に、試験ルートで参照するシンボル（変数）を自動抽出し、また詳細設計チャートを基に、ブレークポイントを自動抽出し、さらに試験ルートで参照するシンボル（変数）とブレークポイントを基に、その試験手順を試験手順書ひな形として、ファイルおよび規定フォーマットに自動出力するものであって、以下の各機能ブロック要素から構成されている。

【0033】・動作環境解析部

- ・試験ルート解析部
- ・書式標準化変換部
- ・構文要素解析部
- ・変数情報収集部
- ・分岐情報収集部
- ・手順書ひな形出力部

【0034】図9は、試験手順書作成機能部の詳細機能の構成例を示すブロック図であって、試験手順書作成機能部メイン20のもとに、動作環境解析部21、試験ルート解析部22、ソース解析部23、手順書ひな形出力部24を有することが示されている。

【0035】図10は、ソース解析部の機能構成例を示すブロック図であって、ソース解析部23のもとに、書式標準化変換部231、構文要素解析部232を有し、さらに構文要素解析部232のもとに変数情報収集部233、分岐情報収集部234を有することが示されている。

【0036】図11、図12は、試験手順書作成機能部の動作フローの概要を示す図(1/2)、(2/2)であって、図中において、◇は判定文を示し、□は処理を示している。

【0037】また、図13、図14は、試験ルートおよび試験内容の生成フローを示す図(1/2)、(2/2)であって、図11、図12に示された試験手順書作成フロー中の、(a)～(e)に示す各機能ブロックにおける処理を、現実的な処理に当てはめた例を示す。図中において、◇は判定文を示し、□は処理を示している。

【0038】以下、試験手順書作成時において、設定値変数または期待値変数として、“変数情報テーブル”に登録するか否かの判定処理について説明する。ここで登録するとは、設定値変数（期待値変数）として、エントリに入れることを意味し、登録した順序を考慮して、設定値変数（期待値変数）として、試験手順書上にF I F O形式で展開する。

【0039】これらの条件を満たしたものを設定値または期待値として自動抽出することによって、人間的な思考を含めた試験手順書を構成することができる。

【0040】(1) 設定値変数として登録する条件として、以下の4つの場合がある。なお、重複した変数は、登録しない。

1. 代入文の右部分にある変数
2. 条件式の変数（条件文の条件式が変数であり、その条件式の以前に該当する変数が一度も使用されていない場合のみ）
3. 静的なポインタ変数など設定条件が必要と判断されるもの
4. 下位関数用設定情報のエントリを表示する／しないを選択可能

【0041】(2) 期待値変数として登録する条件は次のようなものである。なお、この場合、重複した変数を取り除くか、動作させるかは、試験手順書作成機能部の動作環境解析における情報によって、選択可能である。

1. 代入文の左部分にある変数
2. 下位関数用確認情報の変数を登録する／しないも選択可能

【0042】図15は、変数登録のためのフローの一例を示したものである。図中において、◇は判定文を示し、□は処理を示している。

【0043】以下、試験手順書作成時の処理手順の一例を説明する。

A. 処理手順一例

(1) 試験項目に合致したソースを、試験項目ごとに生成する（作業ファイルとして生成する）。

【0044】(2) テストルート表をもとに、試験項目ごとにテストルート上でアクセスしている変数名を抜き出す。

1. 変数名から、その変数の記憶クラスや型名を検索して、設定値用または期待値用情報テーブルに登録する。
2. 抜き出した変数名で、重複している変数名の場合には、情報テーブル内で削除する。構造体メンバ名と単なる変数名とが一致していても、それを同一と見なさない。

【0045】3. 共用体宣言がなされている場合、変数の型を解析して、同一エリアを指すものを削除する。

4. 抜き出した変数名に対応して、日本語名標（日本語コメント）がある場合、その日本語名を日本語定義情報ファイルから検索して、情報テーブルに登録する。

5. 配列変数名は、配列の添字の最大数も情報テーブルに登録する。

【0046】(3) 情報テーブルを基にして、設定値や期待値を手順書ひな形に出力する。

1. 設定値変数用の情報テーブルとしてエントリが登録された変数を、FIFO形式で、初期設定値として、手順書ひな形に出力する。ソースロジックと同様な動きを忠実に検証するために、FIFO形式で設定値処理を行なわせている。

【0047】2. 期待値変数用の情報テーブルとしてエントリが登録された変数を、FIFO形式または任意の形式で、期待値として手順書ひな形に出力する。確認値の重複チェック機能を有効とした場合、下位関数と上位関数との間で重複した期待値については、下位関数部のものを削除またはコメント化して、上位のみで確認することも可能としている。

【0048】3. 上記1., 2. で出力する場合は、被試験関数で使用しているAUTO変数（試験対象となっている関数の中でのみ使用される変数）、またはパラメータ、共用体、ループ中で使用されている変数を、それぞれ区別したマークを付けて、手順書ひな形に出力する。このAUTO変数マークの情報を持った変数（設定できる数値の範囲をユーザに示すため変数の型の表記もしている）に対しては、下位関数用試験手順書取り込み情報において、この変数を無視してコマンドファイルへ展開処理を行ない、下位関数への呼び出しロジックの確認機能を実現している。また、共用体、ループ中で使用されている変数属性を明らかにすることによって、ユーザが設定値としての試験データを記述する際の手助けとしている。

【0049】(4) ポインタ変数への試験時の配慮

1. 被試験関数（モジュール）で使用しているポインタ変数（グローバル変数をポインタ参照している場合のみ）を用いてREAD動作を行なっているポインタ参照命令文を検出した場合、そのポインタ参照の変数に値を設定するために、該当ポインタ変数に値をWRITEしているところにブレークポイントを設定し、ポインタ参照の変数に値を設定させる、処理手順および設定値記述を、手順書ひな形に出力する。

【0050】2. ソース上に存在しやすい、ポインタ関連のバグをみつけ出すために、AUTO変数ポインタを使用している場合には、上記1. と似たREAD動作をしても、ブレークポイントの設定および設定値記述を、手順書ひな形に出力しない。

【0051】(5) テストルート上に、下位関数（サブルーチン）が存在する場合、下位手順書取り込みコマンドを手順書ひな形に出力する（デフォルト動作）。また、この手順書取り込みコマンドが不要であるならば、その行を削除することによって、容易に無効にすることができる。

【0052】(6) テストルート表とテスト項目一覧表とYPSソース（YACチャートで記述されているプログラムまたは設計書）とから、以下の情報を抽出して、手順書ひな形のヘッダ部として出力する。

1. 試験項目名（試験手順書ファイル名）
2. テスト内容および効果確認方法
3. 手順書の作成者名
4. 関数名（モジュール名）
5. 試験手順書作成日

【0053】(7) 指定されたテストルート表を基に、試験項目ごとにテストルートで呼び出している関数内で使用しているAUTO変数やポインタ変数に対するブレーク・ポイントは、被試験モジュールで使用しているブレーク・ポイント数（最大数はそのソース内での1地点最大分岐数による）+1の位置のものを使用し、ブレークポイント・クリアと一組として使い回しをし、数を最小限におさえる。

【0054】(8) 以下の情報を取り出し、動作環境を設定できる機能を持つ。この機能は、試験手順書作成機能部の動作環境解析によって行なわれる。または、統括部でも同様なことが行なえる。

- 【0055】1. 詳細設計書チャートとソース行番号対応情報ファイルの入力元指定
2. 名標情報ファイル（変数名情報ファイル）の入力元指定
 3. 試験手順書のひな形ファイルの出力先指定
 4. 変数型宣言ファイル（ソースベース）の入力元指定
 5. 項目セレクト出力の試験項目一覧表ファイルの入力元指定
 6. 項目セレクト出力のテストルート表ファイルの入力元指定
 7. 設定値用インクルードファイル名の入力元指定
 8. 期待値用インクルードファイル名の入力元指定
 9. ソース言語の指定
 10. 試験手順書を取り込まない関数の指定
 11. 試験手順書を取り込む関数の指定

【0056】B. TP処理異常警告メッセージ

手順書作成時には、ユーザが行なった処理（設定値記入や期待値記入、手順の追加等の試験手順書編集機能部

（TE）、試験手順指示ファイル作成機能部（TF）の出力）に対して、その操作が誤っている場合、その旨の情報をメッセージで知らせることによって、試験手順書作成時の設定値もれ、期待値もれ、手順の記載もれ等を適切に伝えられる。

【0057】試験手順書編集機能部（TE）、試験手順指示ファイル作成機能部（TF）が出力する主なメッセージには、以下のものがある。変数 XXXX の設定値が記述されていません。変数 XXXX の期待値が記述されていません。手順書言語 XXXのパラメータが異常です。

【0058】C. 試験手順書フォーマット

試験手順書は、以下に示すような形式によって出力される。

【0059】図16は、試験手順書のフォーマットの一例を示したものである。図17は、サブ手順書のフォーマットの一例を示したものである。図18、図19は、試験手順書の一例を示す図(1/2)、(2/2)である。

【0060】D. 試験手順書言語

試験手順書は、以下のようなコマンド文法に従って、コマンドとコメントとを使用して記述される。

【0061】(1) コマンド文法

試験手順書においては、ヘッダ部の後に手順部が記述される。この手順部に手順書コマンドを記載することによって、シミュレート・デバッグのコマンドを自動生成することができる。

【0062】手順書コマンドは、以下の記述方法で記載される。

```
<:Command>[:<Paramater>:]△<:Paramater>; △ | [=Value]△
e)△ | [=Value]△
```

【0063】① コマンド文字列を検索する。

② コマンド文字列の後ろの「△」を検出……(1) 第一パラメータ

③ 以降は各コマンドの書式に則ってパラメータを検出する。

④ エンドマークを検出する。

⑤ 次のコマンド文字列を検索する。途中の文字列はすべて無視する。

【0064】(a) 数値の扱い

数値は以下の書式に対応する。すべて半角数字。

- ・10進数 n n n n n n
- ・16進数 0 x n n n n
- ・文字列 “ a a a a ”

【0065】(b) 変数の扱い

変数は以下の書式に対応する。

- ・変数 a a a a
- ・配列変数 a a a [b b b]
- ・構造体変数 a a a [b b b] . c c c [d d d] [e e e]

【0066】(2) コメント

行の先頭に「;」がある行が、コメント行である。

【0067】(3) コマンド

本実施形態において、認識できる手順書コマンドとしては、次の各種のものがある。

- ・ブレークポイント・コマンド
- ・実行コマンド
- ・設定コマンド
- ・確認コマンド
- ・取り込みコマンド
- ・直接コマンド

【0068】以下、各種コマンドについて説明する。

【0069】(a) ブレークポイント・コマンド

被試験プログラムの動作をデータ確認や、データ設定等の目的で、一時停止させるコマンドであって、以下の各種がある。

- ・STOP
- ・CLEAR

【0070】1. STOP (Break Point) , NONSTOP (NonBreak Point) 実行しているプログラムを一時停止させ、またはある地点を通過するか否かを確認するためのコマンドである。

2. CLEAR (Clear Break Point) 設定しているブレークポイントをクリアするコマンドである。

【0071】(b) 実行/停止コマンド

被試験プログラムを実行または停止するコマンドであって、以下の各種がある。

- ・GO
- ・STEP

【0072】1. GO

停止しているプログラム・カウンタから、実行を開始するコマンドである。

2. STEP

停止しているプログラム・カウンタから1行だけ実行するコマンドである。

【0073】(c) 設定コマンド

変数やアドレスで指定したメモリに値を設定するコマンドであって、以下の各種がある。

- ・SET
- ・MEM
- ・FIL

【0074】1. SET (Set Variable)

変数、配列、構造体（共用体など）に対して、値を設定するコマンドである。

2. MEM (Set Memory)

指定したアドレスに、値を設定するコマンドである。

3. FIL (Fill)

指定したアドレスから指定した数だけ、指定した値を設定するコマンドである。

【0075】(d) 確認コマンド

変数やアドレスの値を表示させるコマンドである。このコマンドで表示された値を、結果判定ツールで読み取り、期待値と比較してチェックすることで、OKまたはNGを判定するものであって、以下の各種がある。

- ・CHK
- ・DMF
- ・RET
- ・REG

【0076】1. CHK (Check Variable)

変数、配列、構造体（共用体など）の値を確認するコマンドである。

2. CHK (Check Variable) [拡張]

変数、配列、構造体（共用体など）の値を確認するコマンドである。

3. DMP (Monitor Address)

指定したアドレスの値を確認するコマンドである。

4. DMP (Monitor Address) [拡張]

指定したアドレスの値を確認するコマンドである。

【0077】5. RET (Return Value Disply)

試験手順書の最初を指定するコマンド（関数の復帰値を確認するコマンドである）を直接書くことによって、コマンド・ファイルに展開させるコマンドである。

6. REG (Register)

試験手順書の最初を指定するコマンド（指定したレジスタの値を確認するコマンドである）を直接書くことによって、コマンド・ファイルに展開させるコマンドである。

【0078】(e) 取り込みコマンド

下位関数の手順書や共通初期設定、共通確認の情報を記述した手順書を取り込むためのコマンドであって、以下の各種がある。

- ・ INC
- ・ FILE
- ・ CALL

【0079】1. INC (Include File)

下位関数の手順書をインクルードするコマンドである。

2. FILE

各手順書の共通事項（初期値、期待値、手順）を共通ファイルにすることができる。この共通ファイルの手順書への取り込みを行なうコマンドである。

3. CALL

関数を直接呼び出すためのコマンドである。

【0080】(f) 直接コマンド

シミュレート・デバッグのコマンドを、直接記載するための手順書コマンドであって、次のものがある。

- ・ DR

【0081】1. DR (Direct)

シミュレート・デバッグのコマンドを、直接書くことによって、コマンド・ファイルに展開させるコマンドである。

【0082】(g) 特殊コマンド

特殊コマンドは、以下に示す手順ブロックの範囲を示すために使用されるコマンドである。

- ・ 手順書区分 (START~END)
- ・ 前設定区分 (PSET~PEND)
- ・ 初期設定区分 (INIT~IEND)
- ・ 期待値区分 (EXPC~EEND)
- ・ 停止区分 (STOP~SEND)

これらの区分は、試験手順書作成機能部 (TP) によって手順書に組み込まれる。また試験手順書編集機能部 (TE) によって任意編集が可能である。

【0083】1. 手順書区分

手順書上で、手順書言語を使用できる範囲を指定する。

2. 前設定区分

手順書上で、被試験対象関数を呼び出す前に行なう設定処理記述範囲を指定する。このブロックは、START~ENDの外に記述する。

3. 初期設定区分

手順書上で、初期設定を行なう範囲を指定する。

4. 期待値区分

手順書上で、期待値確認を行なう範囲を指定する。

5. 停止区分

停止地点を指定する。この中に、停止したとき行なう動作（設定、確認）を記述できる。

【0084】(h) その他

その他に、手順書言語として、特殊な意味に用いられるものとして、以下の各種がある。

- ・ 継続行を示す (\$)
- ・ 変数従属性を示す (AT, PT, UN, for)
- ・ 変数型情報 (ss, si, ui, ul)

その変数で設定（確認）できる範囲を示すこれらの区分は、試験手順書作成機能部によって、手順書に組み込まれる。

【0085】1. 継続行表記

手順書上で、手順書言語の1行の範囲を指定する。

2. 変数属性

手順書上で、変数の属性を指定する。

【0086】〔3〕試験手順書編集機能部 (TE)

試験手順書編集機能部における手順書編集は、次のように行なわれる。

【0087】(1) 試験手順書作成機能部で作成された手順書ひな形に、以下に示すデータを入力させる。

1. 設定値
2. 期待値
3. 呼び出す関数の試験項目番号

【0088】(2) 以下に示す手順書作成機能では、自動出力できない手順書コマンド（手順書言語）を必要に応じて追加する。

1. 当該試験項目のルートではアクセスしていない変数（配列や構造体を含む）の確認を追加する場合

- ・ 変数名の入力（日本語、英字名）
- ・ 変数の型の入力
- ・ 設定値または期待値の入力

【0089】2. ルート通過確認を行なうために、各分岐点にブレークポイントを追加設定する場合

- ・ ブレークポイント指定（行番号、関数名指定）
- ・ ノンストップブレークポイント指定（行番号、関数名指定）

【0090】3. while文やfor文の中でアクセスしている変数（配列や構造体を含む）の設定値や期待値を、ループ回数分の変数の確認ができるように、コマンドを追加する場合

- ・変数名の入力（日本語、英字名）

- ・変数の型の入力

- ・設定値または期待値の入力

【0091】4. while文やfor文の中で呼んでいる関数を、ループ回数分アクセスできるように、コマンドを追加する場合

- ・アクセス・ブレークポイントの入力

- ・実行コマンド（GO）の入力

- ・呼び出す関数の試験項目番号の入力

【0092】5. モジュールの途中で関数が終了している所に、ブレークポイントを追加設定する場合

- ・ブレークポイントの行番号指定

6. テーブルジャンプによる関数呼び出し（試験項目に対応する該当関数の呼び出しができるように、コマンドの追加を行なう）

- ・呼び出し関数の試験項目番号の入力

7. 試験用初期値設定を行なうための手順書取り込み

8. 試験結果確認者名の入力

【0093】〔4〕試験手順指示ファイル作成機能部（TF）

試験手順指示ファイル作成機能部は、試験手順書を基に、任意のデバッグに対応する試験用コマンドを自動生成して試験を実行し、試験対象となる上位モジュールが利用する下位モジュールを自動連結して、ボトムアップ試験を行なうとともに、下位モジュールの試験方法（ホワイトボックスまたはブラックボックス）を必要に応じて選択するものであって、以下の各機能ブロック要素から構成されている。

- ・動作環境解析部

- ・下位関数取込み部

- ・手順書言語変換部

- ・試験手順指示ファイル作成部

【0094】図20は、試験手順指示ファイル作成機能部の詳細機能の構成例を示すブロック図であって、試験手順書作成機能部メイン40のもとに、動作環境解析部41、下位関数取込み部42、手順書言語変換部43、試験手順指示ファイル作成部44を有することが示されている。

【0095】図21は、試験手順指示ファイル作成機能部の動作フローの概要を示したものであって、図中、◇は判定文を示し、□は処理を示している。

【0096】以下、試験手順指示ファイル（コマンド・ファイル）作成時の処理手順の一例を説明する。

【0097】(1) 試験手順書編集機能部の作業によって作成された試験手順書ファイルから、指定されたシンボリック・デバッグに適合する実行コマンドに自動変換して、試験手順指示ファイル（コマンド・ファイル）を生成する。試験手順書は、汎用性、利便性のあるTSS用試験手順書言語を使用して記述されており、この手順書言語と、指定デバッグ用のデバッグコマンドとを対応付

けて変換を実施する。

【0098】(2) 手順書に記載されている以下の項目を、コマンド・ファイルにコメントとして出力する。

1. 試験項目名（コマンドファイル名）

2. 手順ファイル生成日

3. 作成者名

4. 変数名に対する日本語のコメント

【0099】(3) 下位関数がある場合には、試験方法をホワイトボックス試験で実施するか、またはブラックボックス試験で実施するかを選択できる。

【0100】1. ホワイトボックス試験指定時には、

① 下位関数（モジュール）用試験手順書で使用している動的な変数の設定値を削除する。

② 下位関数（モジュール）用試験手順書で使用している動的な変数の期待値は残す。

③ 下位関数（モジュール）用試験手順書で使用している静的な変数の設定値は残す。

④ 下位関数（モジュール）用試験手順書で使用している静的な変数の期待値は残す。

【0101】1. ブラックボックス試験指定時には、

① 下位関数（モジュール）用試験手順書で使用している動的な変数の設定値を削除する。

② 下位関数（モジュール）用試験手順書で使用している動的な変数の期待値を削除する。

③ 下位関数（モジュール）用試験手順書で使用している静的な変数の設定値は残す。

④ 下位関数（モジュール）用試験手順書で使用している静的な変数の期待値を削除する。

【0102】この下位関数取込み機能によって、関数単体試験に限らず、関数間の結合試験の試験実施用データの作成も可能となる。また、ボトムアップ試験やトップダウン試験の利用も可能となる。

【0103】〔5〕試験結果判定機能部（TC）

試験結果判定機能部は、試験結果を期待値と比較して、自動判定を行なうものであって、以下の各機能ブロックから構成されている。

- ・動作環境解析部

- ・試験手順書解析部

- ・試験結果ファイル解析部

- ・結果付試験手順書出力部

- ・試験状況表出力部

【0104】図22は、試験結果判定機能部の詳細機能の構成例を示すブロック図であって、試験結果判定機能部メイン50のもとに、動作環境解析部51、試験手順書照合部52、結果付試験手順書出力部53、試験状況出力部54を有することが示されている。

【0105】図23は、試験手順書照合部の機能構成例を示すブロック図であって、試験手順書照合部52のもとに、試験手順書解析部521、試験結果ファイル解析部522を有することが示されている。

【0106】図24は、試験結果判定機能部の動作フローの概要を示す図であって、図中において、◇は判定文を示し、□は処理を示している。

【0107】(1) デバッグ上で試験を実行して生成された試験結果ファイル（ジャーナル・ファイル等）に記録されている試験結果が、期待値と一致しているか否かをチェックし、NGの場合に該当する試験項目の手順書にある期待値欄の横に、NG表示と結果値を自動入力する。

(2) (1) の結果として、ジャーナルファイルから、試験日と試験結果（OK/NG）を自動入力する。

【0108】(3) 確認者欄は作業者が試験結果をチェックし、その承認をした証拠として、エディタで氏名を記入する（この欄が記述されていない場合は、試験完了としない）。

(4) 進捗管理ツールへの連携機能として、日付、試験実施数、NG数、OK数の出力を行なう。

(5) 問題管理ツールへの連携機能として、日付、OKおよびNGとなった項目の試験項目番号を出力する。

【0109】図25、図26、図27は、結果付試験手順書の一例を示す図(1/3)、(2/3)、(3/3)である。

【0110】〔6〕印刷機能部（TR）

印刷機能部は、指定された試験手順書、結果付試験手順書等のファイルを、既存の印刷ツールを起動して、定型印刷する。

【0111】〔7〕テスト支援システム統括部

テスト支援システム統括部は、以下の各機能ブロック要素から構成されている。これら各機能ブロックは、それぞれが独立している。

- ・動作環境解析部
- ・差分情報収集部
- ・リバース変換部
- ・試験状況管理部

【0112】図28は、テスト支援システム統括部の詳細機能の構成例を示すブロック図であって、テスト支援システム統括部7のもとに、動作環境解析部71、差分情報収集部72、リバース変換部73、試験状況管理部74を有することが示されている。

【0113】(1) 動作環境解析部

テスト支援システムの動作環境の設定を行う。なお、この部分で、試験項目抽出機能部1、試験手順書作成機能部2、試験手順書編集機能部3、試験手順指示ファイル作成機能部4、試験結果判定機能部5、印刷機能部6の動作環境の設定を、並立的に行うようにしてもよい。

【0114】(2) 差分情報収集部

再試験等で試験手順書を作成する場合に、以前使用した試験手順書と、今回新たに作成した試験手順書とを比較して、変更点がある部分以外の設定値情報を引き継ぐ。またTS、TP、TF、TCの各機能部における、前回

の状態をバックアップして、復元可能にさせる機能を持つ。

【0115】(3) リバース変換部

試験手順指示ファイルを、試験手順書として生成する機能をいう。指定されたデバッグ種別を把握して、その条件にあったデバッグコマンドを手順書言語に変換する。

【0116】(4) 試験状況管理部

試験結果判定機能部が出力した試験結果状況表をもとにして、試験進捗状況を把握して、そのスケジューリングを行なう。

【0117】図29、図30は、差分情報収集部の動作フローの概要を示す図(1/2)、(2/2)であって、図中、□は処理を示している。また、図31は、リバース変換部の動作フローの概要を示す図であって、図中、□は処理を示している。

【0118】図32、図33、図34は、それぞれテスト支援システムに係わるデータ連携の概要を示す図(1/3)、(2/3)、(3/3)である。図示のように、手順書作成ツール（TP）、手順書編集作業（TE）、コマンドファイル作成ツール（TF）、結果判定ツール（TC）において、関係する各種ファイルが示されている。

【0119】図35は、従来技術と本発明方式との作業内容の違いを説明するものであって、(a)は従来の作業を示し、(b)は本発明のテスト支援システムによる作業を示している。従来の作業における網かけ部は、手作業が必要な部分（人為ミスが入る恐れがある部分）を示し、本発明方式の作業におけるハッチング部は、作業を自動化した部分を示している。

【0120】図36は、ソフトウェア試験における作業工数を示したものであって、従来の試験作業工数と本システムでの試験作業工数とを試験項目ごとに比較して示している。本システムによれば、従来システムの場合と比較して、試験工数が1/2で済むことが示されている。

【0121】図37は、ソフトウェア試験における不具合検出率を示したものであって、従来システムの場合と本システムの場合とを試験項目ごとに比較して示し、図中、CTはコンポートテスト、ITはインテグレーションテスト、STはシステムテストを示している。本システムによれば、従来システムの場合と比較して、比較的早い段階のテストで、不具合を検出できることが示されている。

【0122】

【発明の効果】以上説明したように本発明によれば、ソフトウェア試験の全体の作業を効率化することができる。とともに、試験の比較的早い段階で、ソフトウェアの品質向上を確保できるので、ソフトウェア全体の開発期間を短縮することができる。また、ソフトウェア品質特性を十分に満足するソフトウェアを提供することが可能となる。

【図面の簡単な説明】

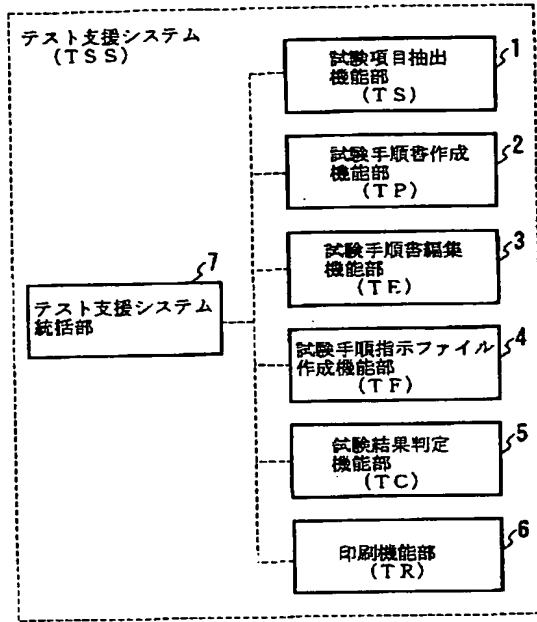
- 【図 1】本発明の原理的構成を示す図である。
- 【図 2】本発明の作用を説明する図(1/2)である。
- 【図 3】本発明の作用を説明する図(2/2)である。
- 【図 4】試験項目抽出機能部の入出力を示す図である。
- 【図 5】試験項目抽出機能部の詳細機能の構成例を示すブロック図である。
- 【図 6】試験項目抽出機能部の動作フローの概略を示す図である。
- 【図 7】試験項目一覧表の出力例を示す図である。
- 【図 8】試験ルート表の出力例を示す図である。
- 【図 9】試験手順書作成機能部の詳細機能の構成例を示すブロック図である。
- 【図 10】ソース解析部の機能構成例を示すブロック図である。
- 【図 11】試験手順書作成機能部の動作フローの概要を示す図(1/2)である。
- 【図 12】試験手順書作成機能部の動作フローの概要を示す図(2/2)である。
- 【図 13】試験ルートおよび試験内容の生成フローを示す図(1/2)である。
- 【図 14】試験ルートおよび試験内容の生成フローを示す図(2/2)である。
- 【図 15】変数登録のためのフローの一例を示す図である。
- 【図 16】試験手順書のフォーマットの一例を示す図である。
- 【図 17】サブ手順書のフォーマットの一例を示す図である。
- 【図 18】試験手順書の一例を示す図(1/2)である。
- 【図 19】試験手順書の一例を示す図(2/2)である。
- 【図 20】試験手順指示ファイル作成機能部の詳細機能の構成例を示すブロック図である。
- 【図 21】試験手順指示ファイル作成機能部の動作フローの概要を示す図である。
- 【図 22】試験結果判定機能部の詳細機能の構成例を示すブロック図である。
- 【図 23】試験手順書照合部の機能構成例を示すブロック図である。
- 【図 24】試験結果判定機能部の動作フローの概要を示す図である。
- 【図 25】結果付試験手順書の一例を示す図(1/3)である。
- 【図 26】結果付試験手順書の一例を示す図(2/3)である。
- 【図 27】結果付試験手順書の一例を示す図(3/3)である。
- 【図 28】テスト支援システム統括部の詳細機能の構成例を示すブロック図である。
- 【図 29】差分情報収集部の動作フローの概要を示す図(1/2)である。
- 【図 30】差分情報収集部の動作フローの概要を示す図(2/2)である。
- 【図 31】リバース変換部の動作フローの概要を示す図である。
- 【図 32】テスト支援システムに係わるデータ連携の概要を示す図(1/3)である。
- 【図 33】テスト支援システムに係わるデータ連携の概要を示す図(2/3)である。
- 【図 34】テスト支援システムに係わるデータ連携の概要を示す図(3/3)である。
- 【図 35】従来技術と本発明方式との作業内容の違いを説明する図であって、(a)は従来の作業を示し、(b)は本発明のテスト支援システムによる作業を示す。
- 【図 36】ソフトウェア試験における作業工数を示す図である。
- 【図 37】ソフトウェア試験における不具合検出率を示す図である。

【符号の説明】

- 1 試験項目抽出機能部 (TS)
- 2 試験手順書作成機能部 (TP)
- 3 試験手順書編集機能部 (TE)
- 4 試験手順指示ファイル作成機能部 (TF)
- 5 試験結果判定機能部 (TC)
- 6 印刷機能部 (TR)
- 7 テスト支援システム統括部

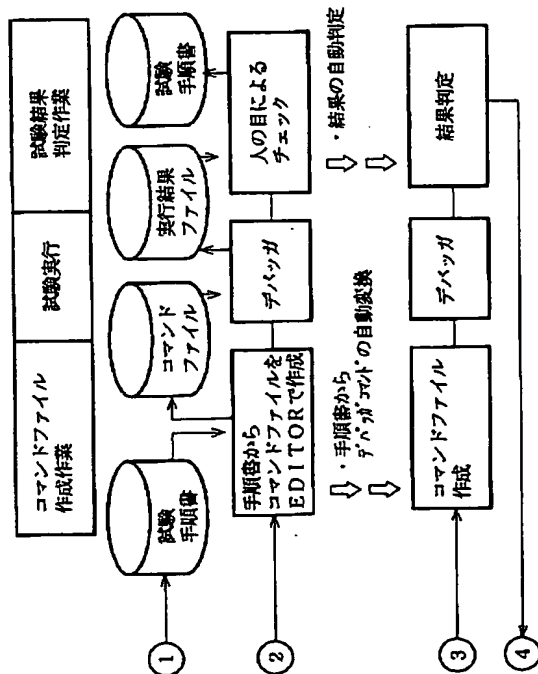
【図1】

本発明の原理的構成を示す図



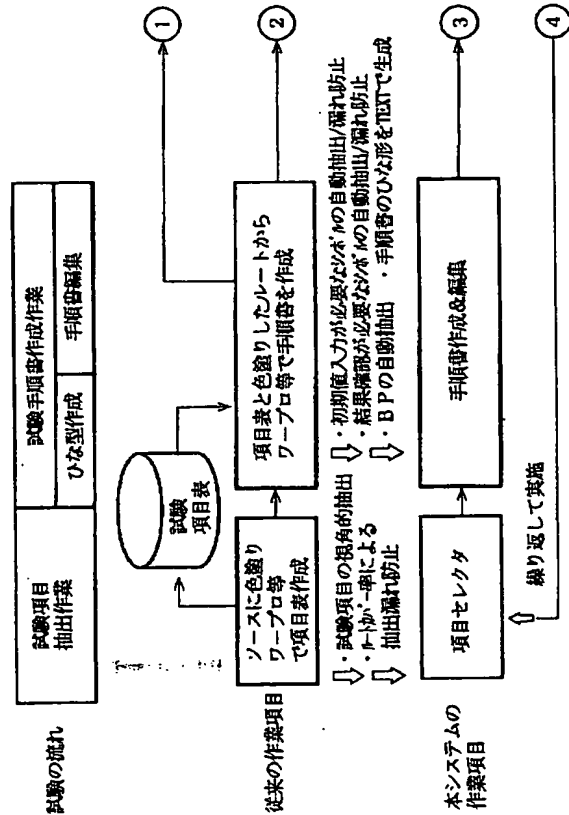
【図3】

本発明の作用を説明する図(2/2)



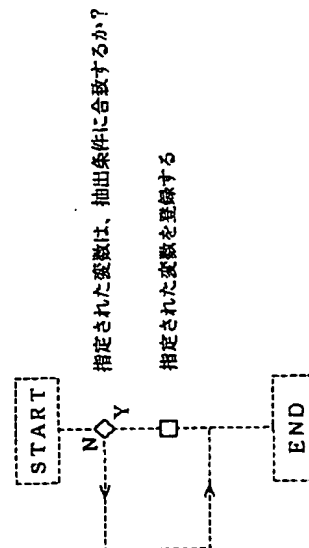
【図2】

本発明の作用を説明する図(1/2)



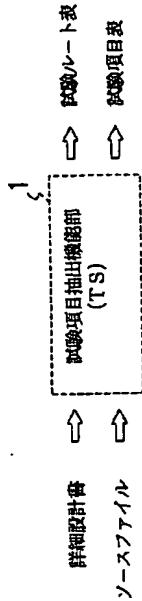
【図15】

変換検査のためのフローの一例を示す図



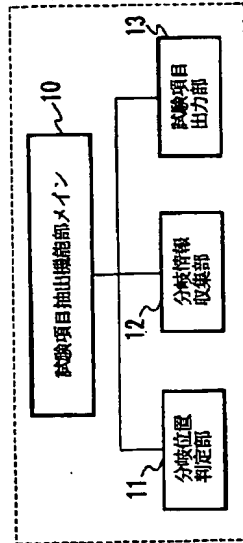
【図 4】

試験項目抽出機能部の入出力を示す図



【図 5】

試験項目抽出機能部の詳細機能の構成例を示すブロック図



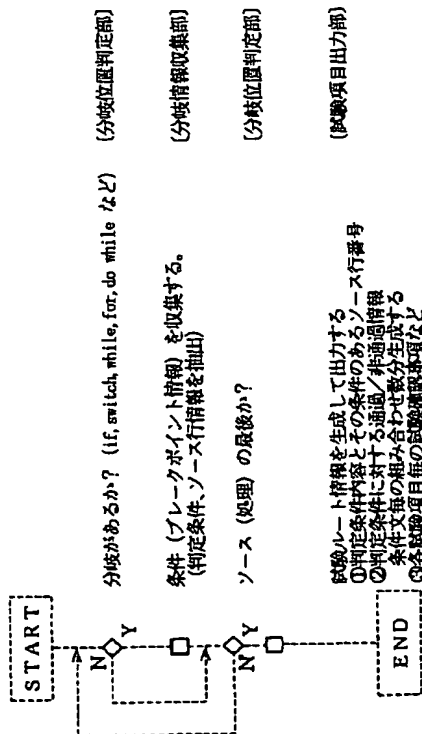
【図 7】

試験項目一覧表の出力例を示す図

UJ_MIRRP	テスト項目一覧表	1994.06.14	備考
No.	テスト項目名 (テスト結果)	テスト内容	テスト名
1	01_01	テスト内容 動作結果が、NGかを判定 (結果確認方法) TSSにより判定する	
2	01_02	テスト内容 動作結果が、OKかを判定 (結果確認方法) TSSにより判定する	

【図 6】

試験項目抽出機能部の動作フローの概略を示す図



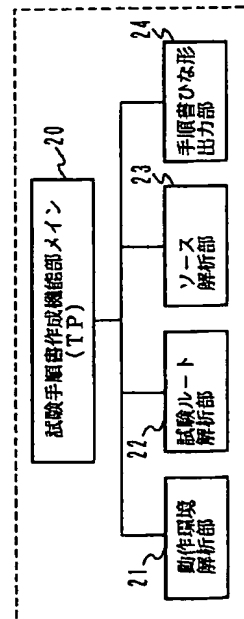
【図 8】

試験レポート表の出力例を示す図

英文字反転プログラム	テストルート表	1994.06.14	藤田 友亮
行番号	条件式	テスト項目番号 / 結果	
0025	判定] 文字配列 (ループカウンタ) が 'a' 以上 'z' 以下なら	真 偽	
0031		真 偽	

【図 9】

試験手順書作成機能部の構成例を示すブロック図

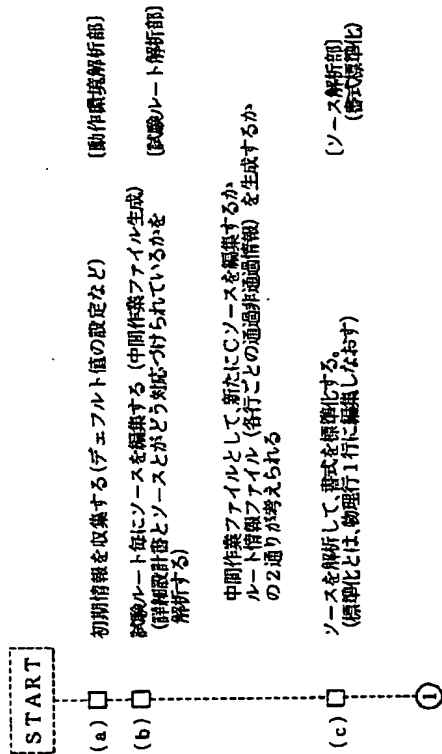
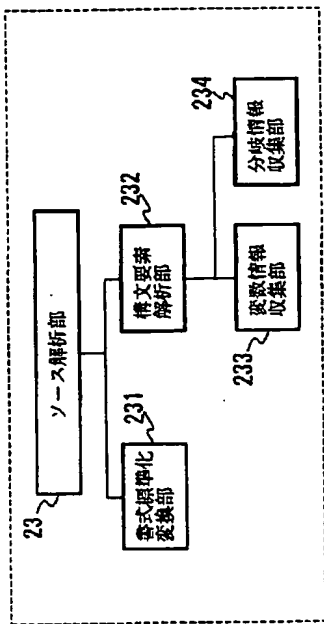


【図10】

【図11】

【図17】

ソース解析部の機能構成例を示す図 試験手順書作成機能部の動作フローの概要を示す図(1/2) サブ手順書のフォーマットの一例を示す図

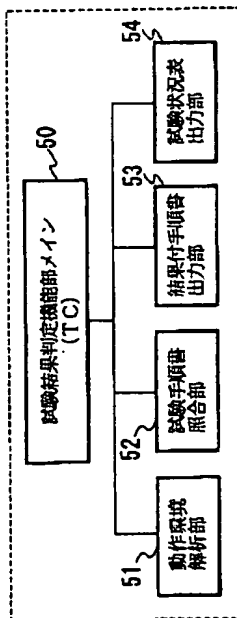
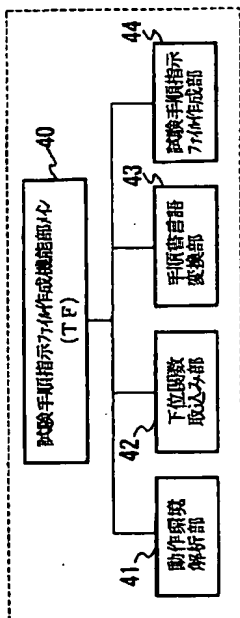


題名	作成者	③	作成日	④
手順	設定値	⑧	期待値	⑨
⑦				

【図20】

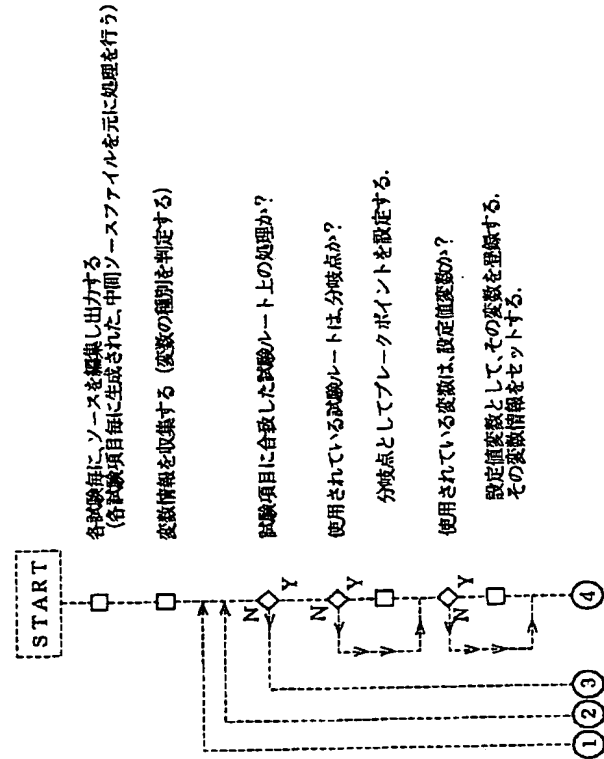
【図22】

試験手順指示ファイル作成機能部の詳細機能の構成例を示すブロック図 試験結果判定機能部の詳細機能の構成例を示すブロック図



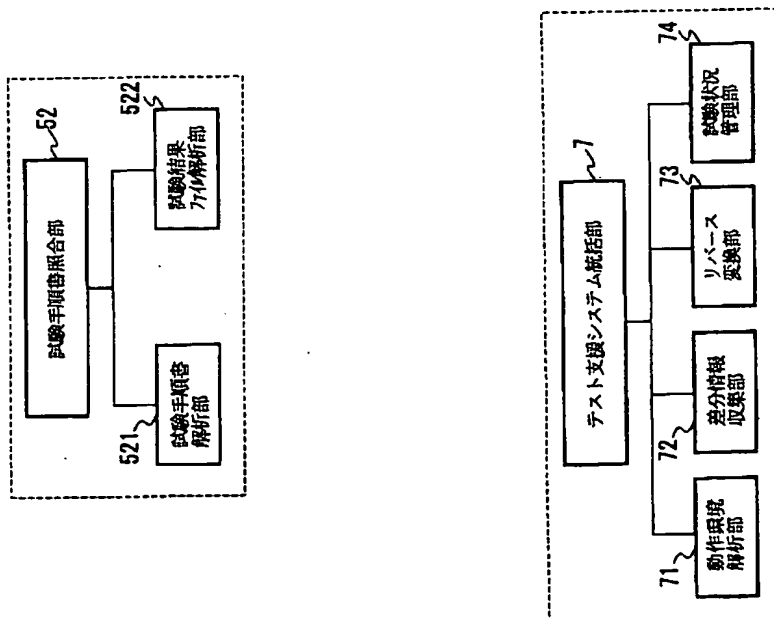
【圖 13】

試験ルートおよび試験内容の生成フローを示す図(1/2)



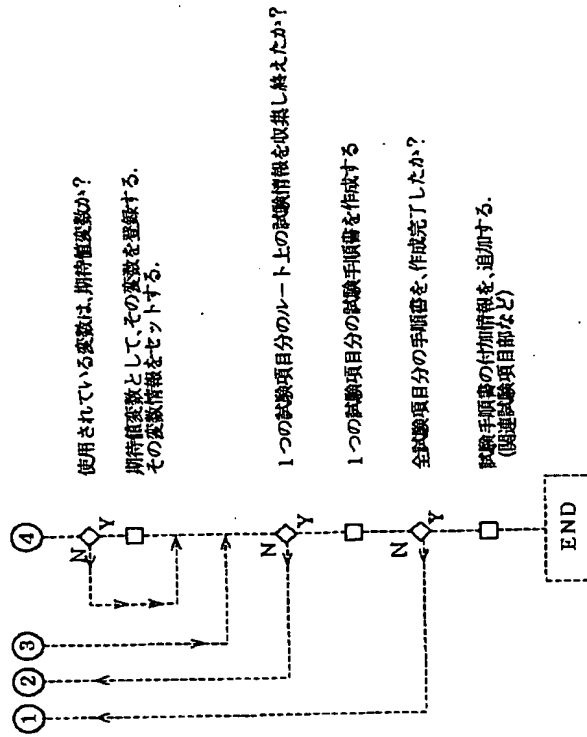
【圖 28】

テスト支援システム統括部の詳細機能の構成例を示すブロック図



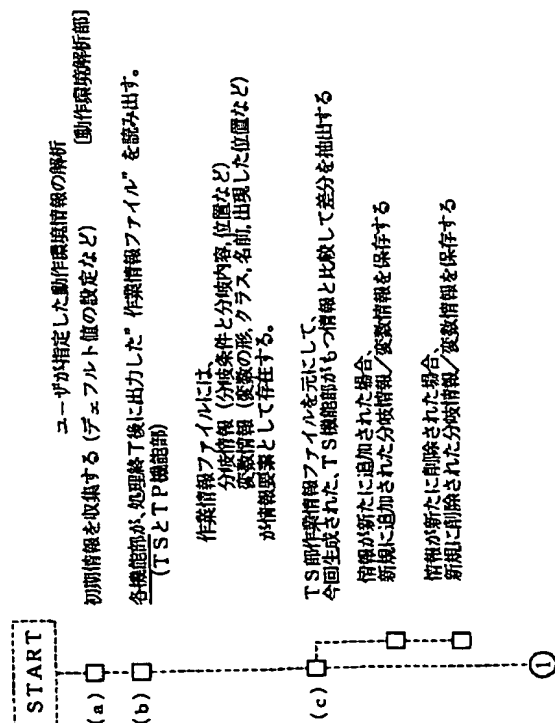
【図 14】

試験ルートおよび試験内容の生成フローを示す図(2/2)



【図 29】

差分情報収集部の動作フローの概観を示す図(1/2)



【図 16】

試験手順書のフォーマットの一例を示す図

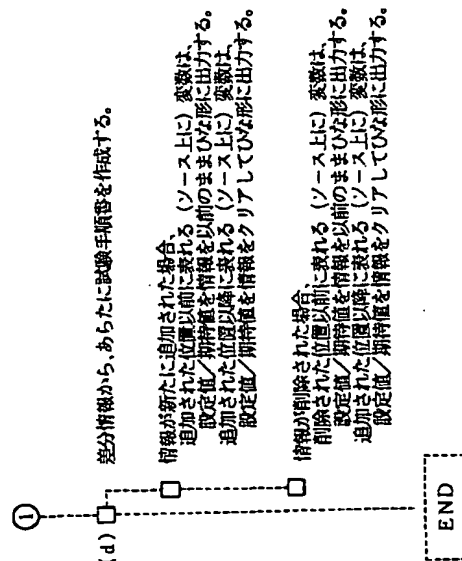
Example format of a test procedure document (Figure 16):

試験バージョン名	①	項目名	②	作成者	③	作成日	④
回	1	2	3	4	5	6	7
試験日							
結果							
確認者							
[テスト内容] ⑤							
[確認試験方法] ⑥							
関連試験項目	1	⑩					
設定ファイル	1	⑪					
確認ファイル	1	⑫					
手順ファイル	1	⑬					
手	順	⑦					
設定値	⑧						
期待値	⑨						

ベッダ部

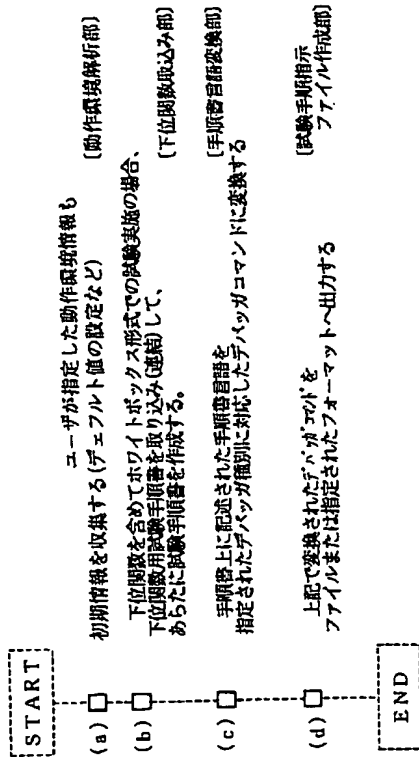
【図 30】

差分情報収集部の動作フローの概観を示す図(2/2)



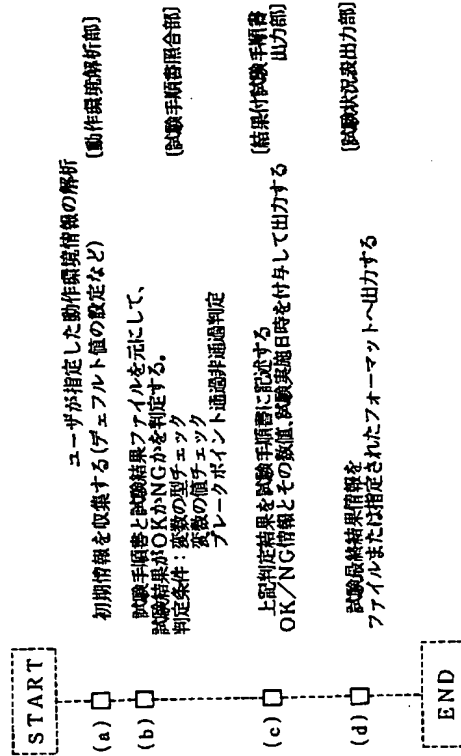
【図21】

試験手順指示ファイル作成機能部の動作フローを示す図



【図24】

試験結果判定機能部の動作フローの概略を示す図



【図 2 7】

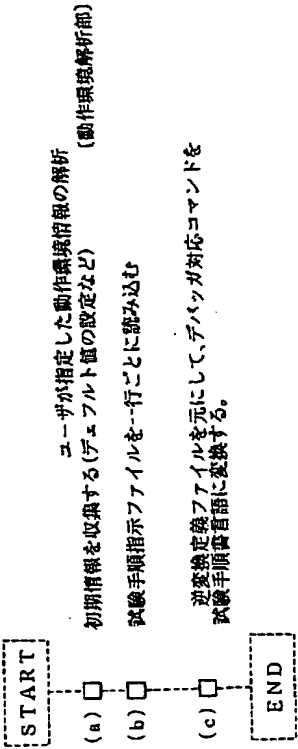
結果付試験手順書の一例を示す図(3/3)

※

【確認】		
EXPC		12/22 14:47 OK
ca_rln	正常	=7
CHK ca_rln		(0x00000007)
www. m1	正常	12/22 14:47 OK
DMP www. m1, 2		=1
		(0x000000001)
		12/22 14:47 OK
		=3
		(0x000000003)
		12/22 14:47 OK
		=7
		(0x000000007)
		12/22 14:47 OK
END		
END		

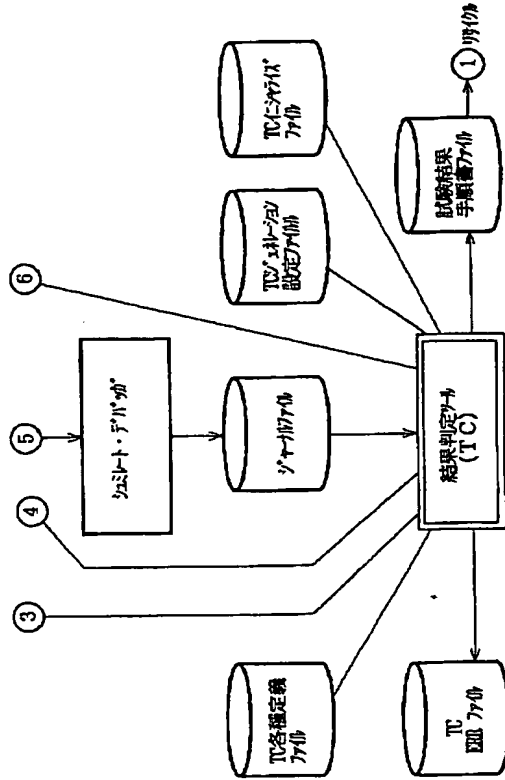
【図 3 1】

リバース交換部の動作フローの概要を示す図



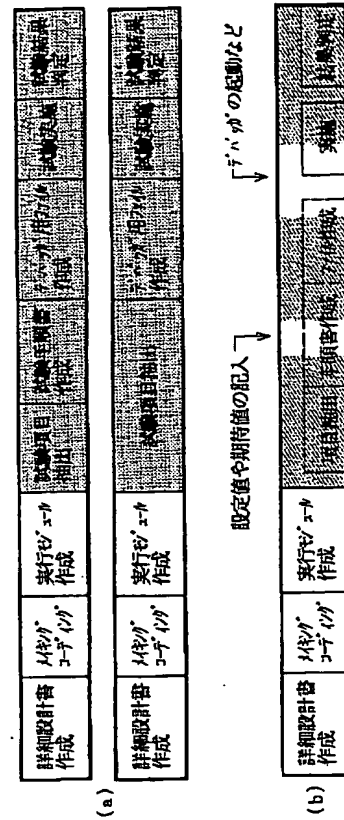
【図34】

テスト支援システムに係わるデータ連携の概要を示す図(3/3)



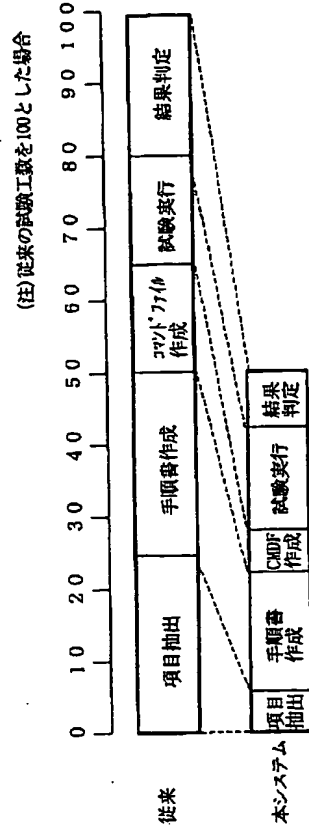
【図35】

従来技術と本発明方式との作業内容の違いを説明する図



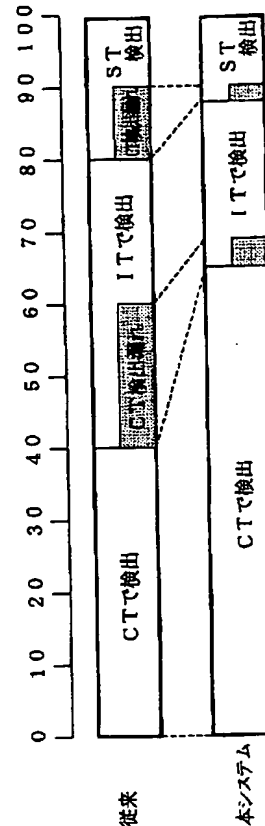
【図36】

ソフトウェア試験における作業工数を示す図



【図37】

ソフトウェア試験における不具合検出率を示す図



フロントページの続き

(72) 発明者 石崎 慎一
 神奈川県横浜市港北区新横浜3丁目9番18
 号 富士通コミュニケーション・システム
 ズ株式会社内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKewed/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.